Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 7, 2019

A. Brotman
Comcast, Inc
S. Farrell
Trinity College Dublin

March 6, 2019

Related Domains By DNS
draft-brotman-rdbd-01

Abstract

This document outlines a mechanism by which a registered domain can publicly document a relationship with a different registered domain, called "Related Domains By DNS", or "RDBD".

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 7, 2019.

Table of Contents

1. Introduction

[[Discussion of this draft is taking place on the dbound@ietf.org mailing list. There's a github repo for this draft at <https://github.com/abrotman/related-domains-by-dns> - issues and PRs are welcome there.]]

Determining relationships between registered domains can be one of the more difficult investigations on the Internet. It is typical to see something such as "example.com" and "dept-example.com" and be unsure if there is an actual relationship between those two domains, or if one might be an attacker attempting to impersonate the other. In some cases, anecdotal evidence from the DNS or WHOIS/RDAP may be sufficient. However, service providers of various kinds may err on the side of caution and treat one of the domains as untrustworthy or abusive because it is not clear that the two domains are in fact related. This specification provides a way for one domain to explicitly document a relationship with another, utilizing DNS records.

Possible use cases include:

- where a company has websites in different languages, and would like to correlate their ownership more easily, consider "example.de" and "example.ie" registered by regional offices of the same company;

- following an acquisition, a domain holder might want to indicate that example.net is now related to example.com in order to make a later migration easier;

- when doing Internet surveys, we should be able to provide more accurate results if we have information as to which domains are related.

It is not a goal of this specification to provide a high-level of assurance that two domains are definitely related, nor to provide fine-grained detail about the kind of relationship that may exist between domains.

Using "Related Domains By DNS", or "RDBD", it is possible to declare that two domains are related.

We include an optional digital signature mechanism that can somewhat improve the level of assurance with which an RDBD declaration can be handled. This mechanism is partly modelled on how DKIM [RFC6376] handles public keys and signatures - a public key is hosted at the relating-domain (e.g., "example.com") and a reference from the related-domain (e.g., "dept-example.com") contains a signature (verifiable with the "example.com" public key) over the text representation ('A-label') of the two domain names (plus a couple of other inputs).

RDBD is intended to demonstrate a relationship between registered domains, not individual hostnames. That is to say that the relationship should exist between "example.com" and "dept-example.com", not "foo.example.com" and "bar.dept-example.com" (where those latter two are hosts).

There already exists Vouch By Reference (VBR) [RFC5518], however this only applies to email. RDBD could be a more general purpose solution that could be applied to other use cases, as well as for SMTP transactions.

This document describes the various options, how to create records, and the method of validation, if the option to use digital signatures is chosen.

1.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The following terms are used throughout this document:

- Relating-domain: this refers to the domain that is declarating a relationship exists. (This was called the "parent/primary" in -00).

- Related-domain: This refers to the domain that is referenced by the relating-domain, such as "dept-example.com". (This was called the "secondary" in -00.)

2. New Resource Record Types

We define two new RRTYPES, an optional one for the relating-domain (RDBDKEY) to store a public key for when signatures are in use and one for use in related-domains (RDBD).

2.1. RDBDKEY Resource Record Definition

The RDBDKEY record is published at the apex of the relating-domain zone.

The wire and presentation format of the RDBDKEY resource record is identical to the DNSKEY record. [RFC4034]

[[All going well, at some point we'll be able to say...]] IANA has allocated RR code TBD for the RDBDKEY resource record via Expert Review.

The RDBDKEY RR uses the same registries as DNSKEY for its fields. (This follows the precedent set for CDNSKEY in [RFC7344].)

No special processing is performed by authoritative servers or by resolvers, when serving or resolving. For all practical purposes, RDBDKEY is a regular RR type.

The flags field of RDBDKEY records MUST be zero. [[Is that correct/ ok? I've no idea really:-)]]

2.2. RDBD Resource Record Definition

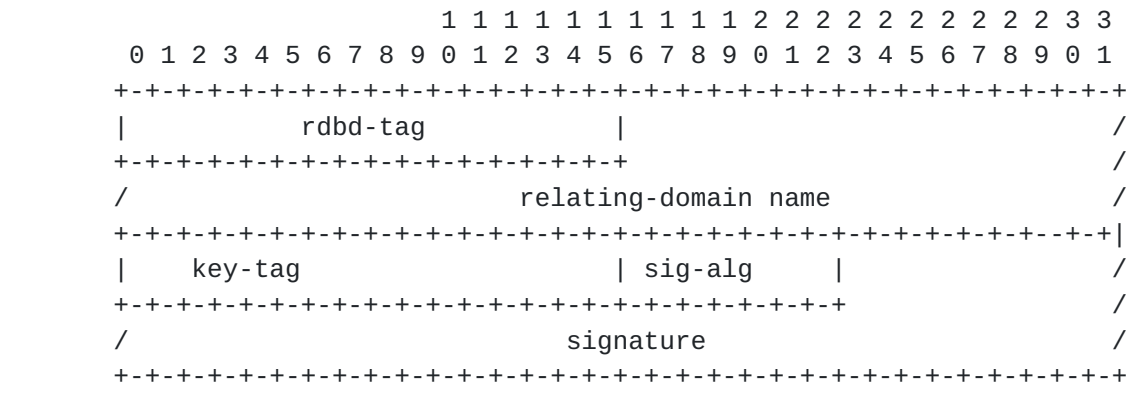
The RDBD resource record is published at the apex of the related-domain zone.

[[All going well, at some point we'll be able to say...]] IANA has allocated RR code TBD for the RDBD resource record via Expert Review.

The RDBD RR is class independent.

The RDBD RR has no special Time to Live (TTL) requirements.

The wire format for an RDBD RDATA consists of a two octet rdbd-tag, the relating-domain name, and the optional signature fields which are: a two-octet key-tag, a one-octet signature algorithm, and the digital signature bits.



The rdbd-tag field MUST contain the value zero. Later specifications can define new rdbd-tag values.

If an optional signture is included, the sig-alg field MUST contain the signature algorithm used, with the same values used as would be used in an RRSIG. The key-tag MUST match the RDBDKEY RR value for the corresponding public key.

If the optional signature is omitted, then the presentation form of the key-tag, sig-alg and signature fields MAY be omitted. If not omitted then the sig-alg and key-tag fields MUST be zero and the signature field MUST be a an empty string. [[Is that the right way to have optional fields in RRs? Not sure.]]

The input to signing ("to-be-signed" data) is the concatenation of the following linefeed-separated (where linefeed has the value '0x0a') lines:

```
relating=<relating-domain>
related=<related-domain>
rdbd-tag=<rdbd-tag value>
key-tag=<key-tag>
sig-alg=<sig-alg>
```

The relating-domain and related-domain values MUST be the 'A-label' representation of these names.

The trailing "." representing the DNS root MUST NOT be included in the to-be-signed data, so a relating-domain value above might be "example.com" but "example.com." MUST NOT be used as input to signing.

A linefeed MUST be included after the "sig-alg" value in the last line.

[[Presentation syntax and to-be-signed details are very liable to change.]]

See the examples in the Appendix for further details.

3. Directionality and Cardinality

RDBD relationships are uni-directional. If bi-directional relationships exist, then both domains can publish RDBD RRs and optionally sign those.

If one domain has relationships with many others, then the relevant RDBD RRs (and RDBDKEY RRs) can be published to represent those.

4. Required Signature Algorithms

Consumers of RDBD RRs MAY support signature verification. They MUST be able to parse/process unsigned or signed RDBD RRs even if they cannot cryptographically verify signatures.

Implementations producing RDBD RRs SHOULD support optional signing of those and production of RDBDKEY RRs.

Implementations of this specification that support signing or verifying signatures MUST support use of RSA with SHA256 (sig-alg==8) with at least 2048 bit RSA keys. [RFC5702]

RSA keys SHOULD use a 2048 bit or longer modulus.

Implementations of this specification that support signing or verifying signatures SHOULD support use of Ed25519 (sig-alg==15). [RFC8080][RFC8032]

5. Validation

A validated signature is solely meant to be additional evidence that the two domains are related. The existence of this relationship is not meant to state that the data from either domain should be considered as more trustworthy.

6. Security Considerations

6.1. Efficiacy of signatures

The optional signature mechanism defined here offers no protection against an active attack if both the RDBD and RDBDKEY values are accessed via an untrusted path.

If the RDBDKEY value has been cached, or is otherwise known via some sufficiently secure mechanism, then the RDBD signature does confirm that the holder of the private key (presumably the relating-domain) considered that the relationship with the related-domain was real at some point in time.

6.2. DNSSEC

RDBD does not require DNSSEC. Without DNSSEC it is possible for an attacker to falsify DNS query responses for someone investigating a relationship. Conversely, an attacker could delete the response that would normally demonstrate the relationship, causing the investigating party to believe there is no link between the two domains. An attacker could also replay an old RDBD value that is actually no longer published in the DNS by the related-domain.

Deploying signed records with DNSSEC should allow for detection of these kinds of attack.

If the relating-domain has DNSSEC deployed, but the related-domain does not, then the optional signature can (in a sense) extend the DNSSEC chain to cover the RDBD RR in the related-domain's zone.

If both domains have DNSSEC deployed, and if the relating-domain public key has been cached, then the the signature mechanism provides additional protection against active attacks involving a parent of one of the domains. Such attacks may in any case be less likely and detectable in many scenarios as they would be generic attacks against DNSSEC-signing (e.g. if a regisgtry injected a bogus DS for a

relating-domain into the registry's signed zone). If the public key from the relevant RDNDKEY RRs is read from the DNS at the same time as a related RDBD RR, then the signature mechanism provided here may provide litle additional value over and above DNSSEC.

6.3. Lookup Loops

It's conceivable that an attacker could create a loop of relationships, such as a.com->b.com->c.com->a.com or similar. This could cause a resource issue for any automated system. A system SHOULD only perform three lookups from the first domain (a.com->b.com->c.com->d.com). The related and relating-domains SHOULD attempt to keep links direct and so that only the fewest number of lookups are needed, but it is understood this may not always be possible.

7. IANA Considerations

This document introduces two new DNS RR types, RDBD and RDBDKEY. [[Codepoints for those are not yet allocated by IANA, nor have codepoints been requested so far.]]

[[New rdbd-tag value handling wll need to be defined if we keep that field. Maybe something like: 0-255: RFC required; 256-1023: reserved; 1024-2047: Private use; 2048-65535: FCFS.]]

8. Acknowledgements

Thanks to all who commented on this on the dbound and other lists, in particular to the following who provided comments that caused us to change the draft: Bob Harold, John Levine, Andrew Sullivan, Suzanne Woolf, and Paul Wouters. (We're not implying any of these fine folks actually like this draft btw, but we did change it because of their comments:-) Apologies to anyone we missed, just let us know and we'll add your name here.

9. Informative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4034] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "Resource Records for the DNS Security Extensions", RFC 4034, DOI 10.17487/RFC4034, March 2005, <https://www.rfc-editor.org/info/rfc4034>.

[RFC5518] Hoffman, P., Levine, J., and A. Hathcock, "Vouch By Reference", RFC 5518, DOI 10.17487/RFC5518, April 2009, <https://www.rfc-editor.org/info/rfc5518>.

[RFC5702] Jansen, J., "Use of SHA-2 Algorithms with RSA in DNSKEY and RRSIG Resource Records for DNSSEC", RFC 5702, DOI 10.17487/RFC5702, October 2009, <https://www.rfc-editor.org/info/rfc5702>.

[RFC6376] Crocker, D., Ed., Hansen, T., Ed., and M. Kucherawy, Ed., "DomainKeys Identified Mail (DKIM) Signatures", STD 76, RFC 6376, DOI 10.17487/RFC6376, September 2011, <https://www.rfc-editor.org/info/rfc6376>.

[RFC7344] Kumari, W., Gudmundsson, O., and G. Barwood, "Automating DNSSEC Delegation Trust Maintenance", RFC 7344, DOI 10.17487/RFC7344, September 2014, <https://www.rfc-editor.org/info/rfc7344>.

[RFC8032] Josefsson, S. and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)", RFC 8032, DOI 10.17487/RFC8032, January 2017, <https://www.rfc-editor.org/info/rfc8032>.

[RFC8080] Sury, O. and R. Edmonds, "Edwards-Curve Digital Security Algorithm (EdDSA) for DNSSEC", RFC 8080, DOI 10.17487/RFC8080, February 2017, <https://www.rfc-editor.org/info/rfc8080>.

Appendix A. Examples

[[TODO: script up generation of all samples - it's not unlikely we mucked up somewhere below when generating 'em partly-manually;-)]]

A.1. Sample Unsigned RDBD RR

When example.com is the relating-domain and dept-example.com is the related-domain, an unsigned RDBD RR would look like this in a zone file:

```
dept-example.com. IN 3600 RDBD 0 example.com.
```

The following is equivalent to tbe above:

```
dept-example.com. IN 3600 RDBD 0 example.com. 0 0 ""
```

A.2. Sample RSA Signature

Appendix C of [RFC6376] has some reference material on how to create a set of keys for use in this type of use case. The RSA key length is recommended to be at least 2048 bits instead of the 1024 recommended in that appendix.

Creation of keys:

```
 $ openssl genrsa -out rsa.private 2048
 $ openssl rsa -in rsa.private -out rsa.public -pubout -outform PEM
```

Sample Key:

```
  rsa.private:
  -----BEGIN RSA PRIVATE KEY-----
  MIIEowIBAAKCAQEA2LNjBAdNAtZOMdd3hlemZF8a0onOcEo5g1KWnKzryDCfH4LZ
  kXOPzAJvz4yKMHW5ykOz9OzGL01GMl8ns8Ly9ztBXc4obY5wnQpl4nbvOdf6vyLy
  7Gqgp+dj6RrycSYJdLitiYapHwRyuKmERlQL6MDWLU9ZSWlqskzLVPgwqtT80xch
  U65HipKkr2luSAySZyyNEf58pRea3D3pBkLy5hCDhr2+6GF2q9lJ9qMopd2P/ZXx
  Hkvzl3TFtX6GjP5LTsb2dy3tED7vbf/EyQfVwrs4495a8OUkOBy7V4YkgKbFYSSk
  GPmhWoPbV7hCQjEAURWLM9J7EUou3U1WIqTj1QIDAQABAoIBAH/eAgwrfq6w4/0X
  Bgk4iQ9q6vnWpQCvW5Z40jRq+MnsnshKPrVL+krIGU/fvt7vaIzIPFTGrf7VWxl3
  +oZg/1sRFPYUItjaluqjaxEhWvHH1saYCb2lAV1x9QtkgjBv4F6GZqfi1MJfro32
  QP36s/hIaVjdHHNsB7BkDgr6VEVIR5y2PmW4aLjHCiqsyDIUM4zRcl4exzw+rst1
  z2seOhhJrnYdc+VnkEg5GKENldZ3tZoY3je/OsfNJtKjpArPRqkP1qve3h3uD+PK
  obZ7BM+xok29Fxf6AgC99eDr9BatTa/a8q7NYMkVRLq/JdOF1XUuDDNd3r93Ae4n
  54qqucECgYEA/Xuct8ALG2/6Kd4Lmm9i055LVxdwB+1wG1JNE1IB+OI+6B8W49po
  vK/fFVHMEV2BoRr4EB58Xxa+oICBImIzTUQXYQnMbDzL2N+X3FrkDSGCPZQy7GzD
  wFdpY3ceNShou1bRt4/hPWLLI35ZXM3yqBJeGhbTUmYVdWrkXTNo2wUCgYEA2tpE
  +bg9iIYUJAg/CEpdWn+8ZxhRnBDziN88Grli+arSWaMWE809GyPaeiplbwywnXRb
  vliskE43CcgstnhRKY8dWB2AQnRESvsJKO8rw/ONSxlWTpFc78xxmmNSvOBs4Srv
  quMc6HTMaetCM5/l0PddCY3/rls9FTESf36RXpECgYEA5AF6mHYwB4AT3/ERMtsa
  ZAuw7Sfx58+V1Z2UItrTV1H7D8RXTKE7MO5plb2796rKXWXq2GTzrnzA/5JXldwL
  FWc4OFsd/AY7vlpxOQ6wr3cCte1GWRAEjFCURZnyHBK7Ejgn8BuFmTfyTXzrWOUP
  bksHRiRd9XJJvxJlU8hYexkCgYBhM9i24THTVVnUtyTn1b+o1lsjnxWAL7c674uO
  gxCGu2w6C8leeiXNzBrZb8Mlk4lOJcQpwtDCNzsSySmy0bWas8ngvRmeam16sAzd
  dX0Gx0HWPSasNrwEddVvMPYqlbNGPv+78quAQ4AW+zqoGzjDm1pjSAJrunJi2yzQ
  G7MNQQKBgCZktECUg2xr8vgVTB586sB7PiHp2j8Wabxh+dMiNUEB7qg4HZdzh8XA
  JXnJnZVQWBL0s10yPg9oITWVBcZ3MqgOqsN1QamN9KjzA46ILtpWptz2q3Nw2Tkl
  m7RBP9R9gM9mnl9/azK7Y5uj11/O3cNJLEIWcraKqydPfvxNyEtP
  -----END RSA PRIVATE KEY-----
```

```
rsa.public:
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2LNjBAdNAtZOMdd3hlem
ZF8a0onOcEo5g1KWnKzryDCfH4LZkXOPzAJvz4yKMHW5ykOz9OzGL01GMl8ns8Ly
9ztBXc4obY5wnQpl4nbvOdf6vyLy7Gqgp+dj6RrycSYJdLitiYapHwRyuKmERlQL
6MDWLU9ZSWlqskzLVPgwqtT80xchU65HipKkr2luSAySZyyNEf58pRea3D3pBkLy
5hCDhr2+6GF2q9lJ9qMopd2P/ZXxHkvzl3TFtX6GjP5LTsb2dy3tED7vbf/EyQfV
wrs4495a8OUkOBy7V4YkgKbFYSSkGPmhWoPbV7hCQjEAURWLM9J7EUou3U1WIqTj
1QIDAQAB
-----END PUBLIC KEY-----
```

To calculate the key-tag as specified in Appendix B of [RFC4034] we used python code from: <https://www.v13.gr/blog/?p=239>

File containing to-be-signed data:

```
$ cat to-be-signed-8.txt
relating=example.com
related=foo-example.com
rdbd-tag=0
key-tag=65498
sig-alg=8
$
$ od -x to-be-signed-8.txt
0000000 6572 616c 6974 676e 653d 6178 706d 656c
0000020 632e 6d6f 720a 6c65 7461 6465 643d 7065
0000040 2d74 7865 6d61 6c70 2e65 6f63 0a6d 6472
0000060 6462 742d 6761 303d 6b0a 7965 742d 6761
0000100 363d 3435 3839 730a 6769 612d 676c 383d
0000120 000a
0000121
```

To sign that file:

```
$ openssl dgst -sha256 -sign rsa.private \
    -out rsa.sig to-be-signed-8.txt
$ od -x rsa.sig
0000000 087c d5c9 375f dcba 9edf ce25 e353 9fb9
0000020 6ef4 ca9f a167 6d91 71bb 7487 5edd fe30
0000040 452e d104 724f f593 009b be3f 6006 ba77
0000060 c1f5 edc6 e207 7ab0 69a1 79bf 18e6 eea3
0000100 3562 6ca4 dc73 22c3 1e35 d15c 44be 5f63
0000120 ac68 f61e ea34 432d 9e12 2325 d48c 2fd9
0000140 330d 1caf 5761 6714 eed2 c7e2 4f71 2c1a
0000160 c35b e45e 833b e343 a8e2 3dbf 1a73 02a8
0000200 c686 7240 aa69 df68 a086 8e3e 2a02 ad57
0000220 32df 0e62 4679 3f4e 8afb 0716 1ad6 4300
0000240 03c6 429f 7b6e bf4d ecae d074 9158 99be
0000260 ab0e 3d49 bb42 a84a 071a b959 2d27 3eea
0000300 c9de 0781 dc5b e205 7708 e50b e485 0cdb
0000320 2fbe adee f521 3b75 9c67 66a8 d217 4f6e
0000340 90da 9423 9d8d e683 7110 4368 f70e 80a2
0000360 3a8c 25f1 3655 44a2 a585 d87d ca99 aac9
0000400
```

The presentation fom of a signed RDBD record (with a 3600 TTL) would be:

```
dept-example.com. 3600 RDBD 0 example.com. 65498 8 (
    hfnhVSlTZMFltG2qU+4vyCPbfSMutxuV8zEyBv7GshOcKMOW
    VLFBK116wRUb7wVgG9TSunXyIuCjDqtidEjWftwVZ8SsXBzo
    tJPMq9HbvZaAnfmx4HxxAMHCpX9QJ2cOK/5VobdZm2eZnXl4
    jd9JsLGuez2wVeiCkwk0x6z/tA61SHmDIFJb5zeKbuvbN14y
    ABaNE88pxoj7EMVQD/nVoag2MqtsiaMS3kbvkYXC3gv25hgM
     mzH+kRXGieaPeYly/ai8OSn3X2bksrffqPuiQsVC03UEm9Vn
    YJgzSjnsvNXvwWJpJ9zyWmVbgdR3/vvUsz2pPvKyJsLT9Knl
    fPNpvg== )
```

The base64 encoded value for the signature can be produced using:

```
$ base64 -w48 rsa.sig
hfnhVSlTZMFltG2qU+4vyCPbfSMutxuV8zEyBv7GshOcKMOW
VLFBK116wRUb7wVgG9TSunXyIuCjDqtidEjWftwVZ8SsXBzo
tJPMq9HbvZaAnfmx4HxxAMHCpX9QJ2cOK/5VobdZm2eZnXl4
jd9JsLGuez2wVeiCkwk0x6z/tA61SHmDIFJb5zeKbuvbN14y
ABaNE88pxoj7EMVQD/nVoag2MqtsiaMS3kbvkYXC3gv25hgM
mzH+kRXGieaPeYly/ai8OSn3X2bksrffqPuiQsVC03UEm9Vn
YJgzSjnsvNXvwWJpJ9zyWmVbgdR3/vvUsz2pPvKyJsLT9Knl
fPNpvg==
```

To verify, with "rsa.sig" containing the above signature:

```
$ openssl dgst -sha256 -verify rsa.public \
    -signature rsa.sig to-be-signed.txt
Verified OK
```

The RDBDKEY RR for this example would be:

```
example.com. 3600 RDBDKEY 0 3 8 (
    LS0tLS1CRUdJTiBQVUJMSUMgS0VZLS0tLS0KTUlJQklqQU5C
    Z2txaGtpRzl3MEJBUUVGQUFPQ0FROEFNSUlCQ2dLQ0FRRUEy
    TE5qQkFkTkF0Wk9NZGQzaGxlbQpaRjhhMG9uT2NFbzVnMUtX
    bkt6cnlEQ2ZINExaa1hPUHpBSnZ6NHlLTUhXNXlrT3o5T3pH
    TDAxR01sOG5zOEx5Cjl6dEJYYzRvYlk1d25RcGw0bmJ2T2Rm
    NnZ5THk3R3FncCtkajZScnljU1lKZExpdGlZYXBId1J5dUtt
    RVJsUUwKNk1EV0xVOVpTV2xxc2t6TFZQZ3dxdFQ4MHhjaFU2
    NUhpcEtrcjJsdVNBeVNaeXlORWY1OHBSZWEzRDNwQmtMeQo1
    aENEaHIyKzZHRjJxOWxKOXFNb3BkMlAvWlh4SGt2emwzVEZ0
    WDZHalA1TFRzYjJkeTN0RUQ3dmJmL0V5UWZWCndyczQ0OTVh
    OE9Va09CeTdWNFlrZ0tiRllTU2tHUG1oV29QYlY3aENRakVB
    VVJXTE05SjdFVW91M1UxV0lxVGoKMVFJREFRQUIKLS0tLS1F
    TkQgUFVCTElDIEtFWS0tLS0tCgo= )
```

A.3. Sample Ed25519 Signature

Since OpenSSL does not yet support Ed25519 singing via its command line tool, we generate our example using the python script below. This uses the python library from Appendix A of [RFC8032].

```
#!/usr/bin/env python3
# CODE_BEGINS
import sys, binascii
from eddsa2 import Ed25519

# secret chosen to be 32 octets funnily enuugh:-)
secret="rdbd-example0001rdbd-example0002".encode('utf-8')
privkey,pubkey = Ed25519.keygen(secret)
msg=open('to-be-signed-15.txt','r').read().encode('utf-8')
signature = Ed25519.sign(privkey, pubkey, msg)

print("private:"+ str(binascii.hexlify(privkey)))
print("public:"+ str(binascii.hexlify(pubkey)))
print("sig:"+ str(binascii.hexlify(signature)))
print("to-be-signed:" + str(msg))

with open("ed25519.sig", "wb") as sigf:
    sigf.write(signature)
with open("ed25519.pub","wb") as pubf:
    pubf.write(pubkey)

# CODE_ENDS
```

The to-be-signed-15.txt file contains:

```
$ cat to-be-signed-15.txt
relating=example.com
related=dept-example.com
rdbd-tag=0
key-tag=35988
sig-alg=15
$
```

The output when the above code is run (with some spacing added) is:

```
$ ./ed25519-signer.py
private:
b'726462642d6578616d706c6530303031
  726462642d6578616d706c6530303032'
public:
b'353fc31e1168c91f0af65d6c26fd441f
  b7df9671a23a746bb3ec86be8d35b648'
sig:
b'466a80ce6377b1e4bec563d85b8d55bd
  4a51a5b91c1c1e46a9c4e22a16557c38
  e85ccc8ac05e6046d0066c2c52b3a174
  20b59af9627840ac312f5ab55e11be07'
to-be-signed:
b'relating=example.com\nrelated=dept-example.com\n
  rdbd-tag=0\nkey-tag=35988\nsig-alg=15\n'
```

The presentation form for an RDBD RR would then be:

```
dept-example.com. 3600 RDBD 0 example.com. 35988 15 (
                    RmqAzmN3seS+xWPYW41VvUpRpbkcHB5G
                    qcTiKhZVfDjoXMyKwF5gRtAGbCxSs6F0
                    ILWa+WJ4QKwxL1q1XhG+Bw== )
```

The RDBDKEY for this example would be:

```
example.com. 3600 RDBDKEY 0 3 15 (
                NT/DHhFoyR8K9l1sJv1EH7fflnGiOnRr
                s+yGvo01tkg= )
```

Appendix B. Changes and Open Issues

[[RFC editor: please delete this appendix]]

B.1. Changes from -00 to -01

- Changed from primary/secondary to relating/related (better suggestions are still welcome)
- Moved away from abuse of TXT RRs
- We now specify optional DNSSEC-like signatures (we'd be fine with moving back to a more DKIM-like mechanism, but wanted to see how this looked)

- Added Ed25519 option
- Re-worked and extended examples

B.2. Open Issues

Current open github issues include:

- #5: specify input for signing more precisely - e.g. is there a CR or NULL or not
- #6: what, if anything, does rdbd for example.com mean for foo.example.com?

These can be seen at: <https://github.com/abrotman/related-domains-by-dns/issues>

Authors' Addresses

Alex Brotman
Comcast, Inc

Email: alex_brotman@comcast.com

Stephen Farrell
Trinity College Dublin

Email: stephen.farrell@cs.tcd.ie